QUESTION-ANSWERING IN A BANKING SYSTEM

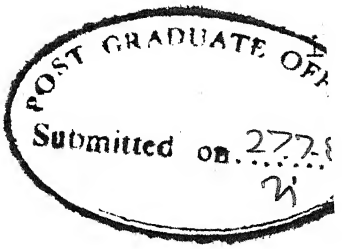
A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By

SHYAM SUNDAR SARKAR

to the
COMPUTER SCIENCE PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JULY, 1983

CSP-1903-M-SAR-QVE



CERTIFICATE

Certified that work presented in this thesis entitled 'Question Answering in a Banking System' by Shyamsundar Sarkar has been carried out under my supervision and it has not been submitted elsewhere for a degree.

July, 1983

Rajeev Sangal

(RAJEEV SANGAL)
Assistant Professor
Computer Science Programme
Indian Institute of Technology
Kanpur

ACKNOWLEDGEMENTS

I would like to thank all those who made this this thesis possible, and in particular, the following:

Rajeev Sangal, for his supervision, his advice and his many ideas that are developed in the thesis.

G.S. Trivedi for typing my thesis in a beautiful way and, finally, my friends, for their encouragements.

SHYAMSUNDAR SARKAR

TABLE OF CONTENTS

ABSTRACT

The thesis deals with question answerning in a banking system. A script of the above domain of knowledge is made using PREP, a powerful knowledge representation system, as a tool. Questions are categorized into nine different types and various retrieval heuristics are designed to get the answers from the banking script. The system can be further extended to deal with several other problems.

CHAPTER 1

INTRODUCTION

Question answering can be viewed as one particular task in human information processing. The idea of human information processing is itself fairly recent. Theories of general information processing are traditionally mathematical theories motivated by problems in computer science. But a growing number of researchers in Computer Science and psychology are now finding it useful to study people in terms of their information processing abilities. It is from this perspective that we can systematically investigate phenomena in question answering.

When we study the process of question answering, we are using methods from computer science. The theories must be associated with sufficient details so that we can implement them by computer programs using various data structures; the main task is to produce a particular memory structure using the information in our domain, so that answers to various questions can be given from the structure. To create such a structure and also for easy retrieval tasks, we need to use an efficient knowledge representation system.

Question answering systems was developed in various places in the past. One such well known system

is QUAIM (Ref.(1)). As such, there is a computer program that is an implementation of QUAIM. This program runs in conjunction with two larger systems, SAM and PAM. Both SAM and PAM are comprehensive story understanding systems. QUAIM is responsible for the question-answering capabilities of SAM and PAM. Also SHRDLU (Winogard, 1972) was well publicized as the first computer program to understand English. Its world knowledge was limited to the blocks world. SHRDLU simulated maniputation of blocks on a table top. If could respond to requests and inquiries.

LSNLIS (Woods, 1972) was a prototype natural language query system designed for accessing a large data base of technical information about the moon rock samples collected during the Apolo 11 mission.

The PLANES system (WALTZ, 1977) is another query system that accesses a large data base of technical information. The data base for PLANES contains information about the naval aircraft maintenance and flight data. Similarly there are many other question answering systems.

In this thesis, we have developed a system that can answer questions in the domain of banking system. A recently developed knowledge representation system called PREP (Ref.(6)) has been used. The powerful pattern matcher in PREP does the matching task needed for retrieval

and the memory subsystem provides access to the concepts representing knowledge in our domain.

In the thesis, Chapter 2 deals with PREP. Chapter 3, Chapter 4 on the background and theories of question answering system construction. Chapter 5 deals with the particular script of banking system. And Chapter 6 on the retrieval heuristics. Chapter 7 gives the conclusions.

CHAPTER 2

INTRODUCTION TO PREP

PREP is a new general purpose knowledge representation system, (Ref.(2)). It has the following fundamental features:

(1) PREP has four distinct interpretation functions. It has a world model, a language L and extensional interpretation function E, time-sensitive relational interpretation function R, time-insensitive relational interpretation function B, and lastly truth value interpretation function V. So for each expression or formula in PREP, the interpretation functions capture distinct semantic points of view.

(2) The language uses set and binary relational theory operators to represent complex relations in the world model in terms of simpler relations. The operators are namely subset, equality, member, intersection, union, intersection, inversion and composition. Since these operators are mathematically well defined, PREP interpretation functions are also precise and well-defined.

(3) PREP has a quote operator which blocks the interpretation of expressions and formulas. The interpretation of a quoted PREP expression is the

expression itself and not the things which the expression denote.

(4) Since PREP interpretation functions are in the PREP world model and there is also a quote operator, it is possible to represent meta-level knowledge in PREP and to reason about it.

(5) Atomic expressions in PREP resemble words in English. The extensional interpretation of an atomic expression in PREP is usually the set of all things that the English word may refer to.

The PREP World Model

The PREP world model consists of abstract objects. The abstract objects include an atomic entity e.g. House-1 or a set of atomic entities (which model a class) e.g. (House-1, House-2, ...) or a set of ordered pairs e.g. ((House-1, red), (House-2, blue), ...). Propositions are also entities in the PREP model but they are referred by PREP formulas; just as other entities in the model may be referred by PREP expressions. Propositions are also in the model which can be referred to without being asserted.

The Interpretation Functions of PREP

The PREP expressions and formulas can be interpreted by four interpretation functions to give objects in the

PREP model. The four interpretation functions are given below:

R : Time-sensitive relational interpretation function that will give triples whose third quantity is a moment of time, e.g. (SKY, BLUE, MORNING).

B : Time-insensitive relational interpretation function that will give ordered pairs e.g. (HOUSE,WHITE).

E : Extentional interpretation function that will give a set of objects that the entity interpretated, refers to e.g. (RED, BLUE, GREEN, ...). This set can be obtained by extentional interpretation of the entity'colour'.

V : Truth value interpretation function. This interprets a formula in PREP and gives either TRUE or FLASE, e.g. V (BLUE COLOUR) = TRUE because

$E[\![\text{BLUE}]\!] = \{\text{BLUE}\}$

$E[\![\text{COLOUR}]\!] = \{\text{RED, GREEN, BLUE, ...}\}$.

Now the set and relational operators that are defined in PREP also, are mathematically well-defined. Hence the interpretation functions are also precise and well defined in PREP. The B interpretation function and E interpretation function are related to each other as

$$E[\![\alpha]\!] = \text{Range}\ (B[\![\alpha]\!])$$

Some examples can be cited as:

$$E[\![\text{Father}]\!] = \{\text{John, Henry, Ram,....}\}.$$

All the people in this world who are fathers.

$$B[\![\text{Father}]\!] = \{(\text{Ram, Father of Ram}),.....\}$$

Now the interpretation functions are partial i.e. they are not defined for all sorts of expressions and formulas. Atleast one interpretation function is defined for an expression to be meaningful. Again the interpretation functions are recursive i.e. a complex expression is interpreted in terms of the interpretations of the sub-expressions.

The PREP Language

The PREP language uses some facilities to express complex expressions in terms of simple expressions. A word in PREP is the simplest expression.

Now extensional interpretation of word expressions refers to a set of elements which can be referred to by the word expression; e.g.

$$E[\![\text{Author}]\!] = \{\text{John, Ram,.....}\}.$$

where all the elements in the set are authors. Similarly B interpretation of the word expression Author is given as

$$B[\![\text{Author}]\!] = \{(\text{Book-1, Author of Book-1}),....\}.$$

i.e. a set of binary relations.

In general $E[\![\alpha]\!] = \text{Range}(B[\![\alpha]\!])$. We now define the compound expressions.

(1) Inversion

Let α be an expression. Then α^{-1} is defined in terms of the following interpretations;

$$E[\![\alpha^{-1}]\!] = \{ d \mid < d, r > \in B[\![\alpha]\!] \}$$

$$B[\![\alpha^{-1}]\!] = \{ < r,d > \mid < d,r > \in B[\![\alpha]\!] \}$$

The inverse operator inverts a relation. A relation usually denotes an entity and the property or value of the entity . Inverse operator refers to all those entities that have particular properties. e.g.

$E[\![\text{Colour}^{-1}]\!]$ = The set of all things that have colou

$E[\![\text{subject}^{-1}]\!]$ = The set of all things that have subje

(2) Range Restriction

Let α and β be two expressions in PREP then range restriction is represented as $(\alpha|\beta)$. This is interpreted as

$$B[\![\alpha|\beta]\!] = \{ <d,r> \in B[\![\alpha]\!] \mid r \in E[\![\beta]\!] \} \text{ and}$$

$$E[\![\alpha|\beta]\!] = E[\![\alpha]\!] \cap E[\![\beta]\!]$$

Range restriction defines subclasses or sub-relations in terms of larger classes, e.g. (Author | german) this expression may be extensionally interpreted to give a set of all german authors.

(3) Juxtaposition

Let α and β be two expressions in PREP then juxtaposition $(\alpha\ \beta)$ is interpreted as $E[\![\alpha\ \beta]\!] = \{r \mid \exists x[\langle x,r\rangle \in B[\![\alpha]\!]$ and $x \in E[\![\beta]\!]]\}$

And $B[\![\alpha\ \beta]\!] = \{\langle d,r\rangle \mid \exists x[\langle x,r\rangle \in B[\![\alpha]\!] \langle d,x\rangle \in B[\![\beta]\!]]\}$

Juxtaposition corresponds to composition of relations in the model. Thus for example

$E[\![$ Colour hair $]\!]$ is obtained by the composition of two relations $\{\langle$ thing, colour of a thing $\rangle$, ...$\}$ and $\{\langle$ thing , thing's hair $\rangle$, ... $\}$.

Thus we can make a correspondence among these operators and the binary theory operators. For example inversion corresponds to inverse of a relation. Range restriction corresponds to intersection. Juxtaposition corresponds to composition of relations and so on.

Propositions are also there in PREP and these are interpreted by truth value interpretation function. Thus this function interpretes a proposition to give either TRUE or FALSE.

Now there are three types of atomic propositions namely, subset, member and equality.

(a) Subset

$V[\![\alpha \subseteq \beta]\!]$ = TRUE iff

$E[\![\alpha]\!] \subseteq E[\![\beta]\!]$ e.g. (Apple $\subseteq$ fruit).

(b) Membership

$V[\![\alpha \in \beta]\!]$ = TRUE iff

$E[\![\alpha]\!] \subseteq E[\![\beta]\!]$ where $E[\![\alpha]\!]$ is singular set.

e.g. (Apple-47 $\in$ Apple).

(c) Equality

$V[\![\alpha = \beta]\!]$ = TRUE iff

$E[\![\alpha]\!] \subseteq E[\![\beta]\!]$ and $E[\![\beta]\!] \subseteq E[\![\alpha]\!]$.

Now we can construct axioms of PREP as

$(\alpha\ \beta : \alpha)$ e.g. (Colour ship : Colour)

Similarly $(\alpha|\beta : \alpha)$ e.g. $(\text{Ship}|\text{Colour}^{-1} : \text{ship})$.

A basic inference rule in PREP is

$$\alpha : \beta\ ,\ \beta : \gamma \vdash \alpha : \gamma$$

An inference rule is defined to be sound iff the truth value interpretation of the consequence is true iff the truth value interpretation of all the premises are true.

Inference Rule

$$\alpha : \beta\ ,\ \beta : \gamma \vdash a : \gamma$$

Proof of Soundness

Assumming $V[\![\alpha : \beta]\!]$ = TRUE and

$V[\![\beta : \gamma]\!]$ = TRUE

then to show that $V[\![\alpha : \gamma]\!]$ = TRUE .

1. $E[\![\alpha]\!] \subseteq E[\![\beta]\!]$ and $E[\![\beta]\!] \subseteq E[\![\gamma]\!]$

this is from the definition of V.

2. $E[\![\alpha]\!] \subseteq E[\![\gamma]\!]$

because the subset operator in set theory is transitive.

3. $V[\![\alpha : \gamma]\!]$ = TRUE from the definition of V.

(':' has the meaning of subset operator in the implementation).

Implementation of PREP

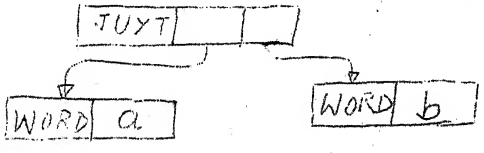
Implementation of PREP involves different stages. Two basic stages which are independent of the semantics of PREP will be described here in short. These two stages are namely (1) Memory subsystem and (2) Pattern matcher.

Stage 1 : Memory subsystem

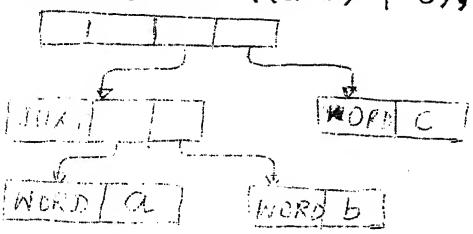
Goal of this part of implementation is to create a suitable data structure to store the expressions and formulas in PREP. Thus to get a concept in memory is the first thing. There are three sub-stages just before memory subsystem namely(a) Notation reader, (b) Parser,(c) Absorber.

Thus when a PREP expression is given notation reader will read it and will produce certain changes before passing it to the parser. The parser will take the output of notation reader and will produce an intermediate form (something like a tree structure).

e.g. (a b) will be sent to the parser to produce an intermediate structure of the form



Similarly for ((a b) | c), the parser will generate:



Next these outputs from the parser will go to the absorber which will call the memory subsystem functions to generate the final concept in memory. A word expression 'a' will be finally generated as

Word	F1
Flags	F2
a	F3
Nil	F4
List of Imm-user	F5

Thus the above structure has five fields , F1, F2, F3, F4,F5. The fields contain

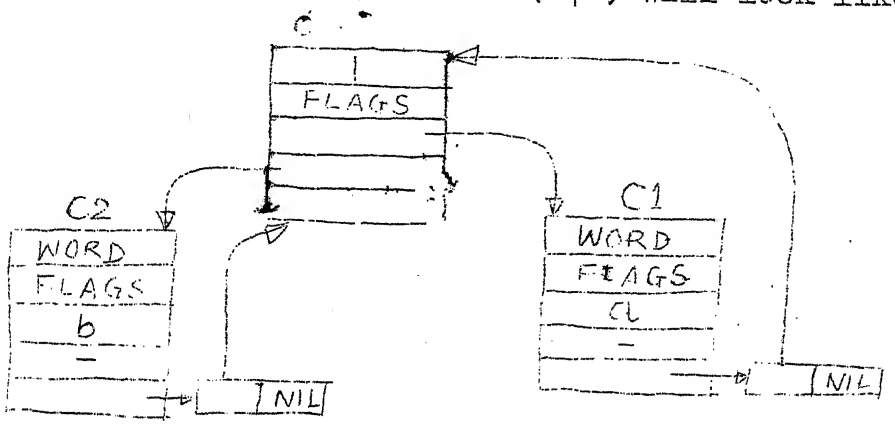
F1 : type of expression

F2 : **flags**

F3 : concepts corresponding to the left hand side expression in case of binary expression or the expression itself in case of unary expression or the list of concepts corresponding to the sub-expressions in the nary expression or LISP atom corresponding to the word in case of word expressions.

F4 : right hand side expression in case of binary operators and nil otherwise.

F5 : this is the pointer to a list which is called the Imm-uses list. This is the list of all expressions for which the current expression is an immediate constituent. Thus in memory the expression (a|b) will look like:



Again in the property-list of the word 'a' we put the concept C1. Hence we can easily get this concept from the property list of 'a'.

State 2 : Pattern Matching

An expression having one or more variable is called a pattern. Now if we give a particular pattern representing a PREP expression, we may need to find whether an expression is there in the memory system that will match the above mentioned pattern. This task is done by pattern matcher. The given pattern may be having a variable in it. So the pattern matcher will generate a list of bindings along with the matched pattern. Each binding gives a variable

and the value of the variable. Thus the list may be called an A-list (i.e. association list).

In each pattern, there is a flag showing whether the pattern consists of only variables. If so, the matcher will not consider this pattern for matching. In a general pattern, the matcher considers the first word concept it comes across and then from the imm-uses list of the concept, it finds out all the patterns which use that particular word concept. Then using those subset of patterns it continues with its matching process and generates a stream of matched pairs (Ref.(4)).

Using these two basic stages of PREP-body we can implement many things like production system, inference or question - answering etc.

CHAPTER 3

QUESTION AND ANSWER

Background

We can visualise question-answering as a process which involves various steps and stages. Before describing the various stages involved in question answering, some discussion is made on conceptual dependency theory. This theory defines primative acts and events in the real world are represented by sequences of primative acts(Ref.(3)).

Conceptual dependency deals with the meaning of a sentence and breaks it into a set of primitive actions. When two sentences are identical in meaning, their conceptual dependency representation will be the same. Thus for example 'John kicked the ball' and 'John hit the ball with foot' will have the same conceptual dependency representation.

Cognitive process depends on the meaning of the sentence and not on the lexical expression of the sentence. Thus for example, the two sentences 'Ram bought a book from Shyam' and 'Shyam sold a book to Ram' have the same meaning and should have the same conceptual dependency representation. Thus a question like 'Did Ram buy the book from Shyam?' Can be answered from any one of the above two sentences.

Conceptual-dependency representation depends on the following acts which are proved to be sufficient for the general

world -knowledge.

ATRANS: Transfer of possession, ownership or control. ATRANS involves an agent, an object, source and recipient. Thus Ram gives a book to Shyam is an ATRANS with agent (Ram), object (book), source (Ram), recipient (Shyam).

PTRANS: The transfer of a physical location. This action involves agent object, destination. Thus 'Ram goes to Delhi' is a PTRANS with agent (Ram), object (Ram) , and destination (Delhi).

PROPEL: Application of physical force. This involves physical movement of a part of a body. Thus 'Push', 'Pull', 'Kick' etc. are the actions involving PROPEL. This action involves agent, object, origin and direction.

MTRANS: The transfer of information. An MTRANS can occur between two animals or within the mind of a man. The human memory is partitioned essentially in three parts namely CP (conscious processor which keeps informations of which we are conscious). IM is the intermediate memory which keeps informations which can be readily used by CP and the LTM (long term memory) which stores informations permanently. Now MTRANS involves agent, object, source and recipient. Thus 'tell' is a MTRANS where source and recipients are both human whereas remember is a MTRANS where source and recipient are two mental stages namely LTM and CP.

MBUILD: The thought process that constructs new information from old. Thus the process takes place in IM and takes information from CP giving back again to CP. Examples are 'decide', 'conclude', 'imagine' etc.

Ingest: The internalization of an external object into an animals body. Thus 'eat', 'drink' etc. are all examples of the action Ingest. This involves an agent, object, origin and destination.

EXPEL: The act of pushing an object out of an animals body. Thus excretion, secretion are instances of EXPEL. 'Sweat', 'Spit', etc. are EXPEL .

MOVE: The movement of an animal involving some body part. MOVE is an instrumental to actions like 'Kick', 'hand', 'throw' etc.

SPEAK: Any vocal act. Humans usually perform speaking as an instrumental of MTRANSing.

ATTEND: The act of focussing some sense organ toward some stimulus. ATTEND is always an instrumental to MTRANSing. Thus 'see' is an MTRANSing from eye to CP with an instrumental ATTEND of the eyes to same object.

GRASP: The act of securing contact with an object. Thus 'grab', 'throw' etc. are examples of GRASP.

Causal Chain Construction

Individual conceptualisations can be causally linked together . In a fully expanded causal chain, there

are events and states alternately occurring. And these are connected by causal relations to form a fully expanded causal chain. There are six basic causal links:

Result(r): An event 'results' in a state. This state is different from a mental state.

Reason (R) : Mental activity (MBUILD) may be the 'reason' for performing an action. This link joins mental events with nonmental actions.

Initiate (I) : A state or event can initiate a thought process (MBUILD).

Enable (E) : A state 'enables' an event to occur.

Leadto (L) : This causal link is used to connect two events in a causal chain representation that is not fully expanded. Thus 'L' link shows that there is a causal chain between two events which is not fully shown.

Cancause (C) : This is similar to LEADTO with a modification. The events and states joined by a cancoause link are hypothetical.

Similarly abbreviated links like I/R can be used to replace a causal chain. These are used for contractions.

Introduction to Script Structures

Frequently occurring sequences of events in a given domain are assembled together to form a single entity called a script. Various types of information regarding a script can be kept independent of the causal chains corresponding to

sequence of events in the script. These can also be used for answering questions. Also summaries are kept which give in short the essential thing in the script. The ability to extract the most important aspects of a script instantiation is crucial in many retrieval tasks. There may be several structural forms in scripts namely, linear, trees and so on.

CHAPTER 4

QUESTION CATEGORIEZATION

Questions are to be first understood and then should be answered following different heuristics.

Now essentially there are four stages before the retrieval task can be made to begin. These stages are

(a) Conceptual parse

(b) Memory internalisation

(c) Conceptual categorization and

(d) Inferential analysis.

The first interpretive process i.e. conceptual parse is only language dependent and all other stages are language independent and within conceptual dependency.

In our case we take questions to be in the PREP expression forms and so the conceptual parse is not implemented. Also we do not consider inferential analysis. Only the two stages memory internalization and conceptual categorization will be described.

Memory Internalization

Since the questions are in PREP expression forms, they will be parsed and absorbed to give concepts in memory. How the expressions are absorbed, has been already described.

Conceptual Categorization

Questions can be conceptually categoriezed into thirteen different categories according to Lehnert(Ref.(7)). Among them we consider only nine categories which apply to our domain of knowledge. The categories are

(1) Causal antecedent
(2) Goal orientation
(3) Enablement
(4) Causal consequent
(5) Verification
(6) Disjunctive
(7) Instrumental/procedural
(8) Concept completion
(9) Quantification.

We shall consider each form and will see the PREP forms.

(1) Causal Antecedent

Causal antecedent questions ask about states or events that have in some way caused the concept in question. e.g. Why did Ram go to Delhi ? Its PREP form will be as:

[Ptrans| agent $^{-1}$ Ram | Obj Ram | dest $^{-1}$ Delhi
[* : event][* : leadto x #?]]

Recognizing the Question

All the causal antecedent questions are recognized by finding the following things:

(a) there is a causal chain between two conceptualizations,

(b) the causal link is LEADTO,

(c) all or part of the first conceptualisation is unknown.

Finding the Question Concept

The question concept of a causal antecedent question is found by selecting the consequent conceptualization. Thus in the above question the question concept is

$$[\text{Ptrans} \mid \text{agent}^{-1}\ \text{Ram} \mid \text{obj}^{-1}\ \text{Ram} \mid \text{dest}^{-1}\ \text{Delhi}]$$

(2) Goal Orientation

This question involves a reason link. The questions are usually regarding a mental action which causes a nonmental action to occur. The questions are of the form:

For what purpose did Ram take the book ?

In PREP - expression form this will look like

$$[\text{ATRANS} \mid \text{agent}^{-1}\ \text{Ram} \mid \text{obj}^{-1}\ \text{book} \mid \text{dest}^{-1}\ \text{Ram}\ [\ * : \text{event}][\ * : \text{Reason X} \quad \# ?]]$$

Recognizing the Question: Such questions are recognized by finding (a) Two causally linked conceptualizations. (b) The causal link is 'Reason'. (c) All or part of the first conceptualization is unknown.

Finding the Question-concept: The question concept is found by deleting the first conceptualization. Thus for the above question, the question concept is

$$[\text{ATRANS} \mid \text{Agent}^{-1}\ \text{Ram} \mid \text{Obj}^{-1}\ \text{book} \mid \text{dest}^{-1}\ \text{Ram}\ ].$$

(3) Enablement:

This type of questions also consist of two conceptualisations which are causally linked and the causal link is 'Enable'. The concept in the question is enabled by some unknown act or state. The example of such a question is

'How was Ram able to eat ?'.

The PREP- expression will be

[Ingest | agent^{-1} Ram | Obj^{-1} food

[* : event] [* : Enable (X # ?)]]

The question is recognized by finding (a) Two causally linked conceptualizations. (b) The causal link is 'Enable'. (c) The first state or action is partly or fully unknown.

The question concept is found by deleting the first conceptualization.

(4) Causal Consequent

This type of question is just reverse of causal antecedent type questions. Here also the causal link is LEADTO. But the second concept is unknown.

Example: 'What happened when Ram went to Delhi?'

The PREP-expression is

[Ptrans | agent^{-1} Ram | Obj^{-1} Ram | dest^{-1} Delhi

[* : event] [X # ? : leadto *]]

The question is recognized by the same way as causal antecedent question except that here partly or fully the second conceptualization is unknown.

The question concept is found by deleting the second conceptualisation. The question concept for the above question is [Ptrans | agent^{-1} Ram | Obj^{-1} Ram | dest^{-1} Delhi] .

(5) Verification:

The verification type of questions refer to the truth of an event. This type of questions is answered by either 'yes' or 'no'. The PREP- expression is of the form (action-1 : event). The question is recognized by finding that it is a single conceptualization and all of it is known. The question concept is the action-1. Example of such a question is [(Ptrans | agent^{-1} Ram | Obj^{-1} Ram | source^{-1} Delhi |dest^{-1} Kanpur) : event]

(6) Disjunctive

Disjunctive questions resemble very much the verification type questions. The difference is that there may be two or more than two verification type questions connected by 'OR's.

Here the example of such a question can be given as

'Is Ram going or coming ?' .

In PREP form the thing will look like

[(action-1 : event) V (action-2 : event)] .

Recognizing such a question is quite straight-forward. The question concept is the list of conceptualisations without the OR connectives.

(7) Instrumental/Procedural

Instrumental/procedural questions have totally or partially unknown instrumentality. Examples of such questions

are 'How did Ram go to New York ?' . In PREP- expression form this will look like [Ptrans|agent^{-1} Ram |Obj^{-1}Ram |dest^{-1} (New-York)

[* : event] [means * = X # ?]]

Thus there is an unknown instrumentality. Sometimes instrumentality for an act involves a long sequence of acts rather than a single act. In this case, the unknown instrument is more appropriately described as a procedure.

Examples: 'How did Ram get Shyam's house ?'

In PREP - form

[Ptrans|agent^{-1} Ram |obj^{-1} Ram |dest^{-1}(house Shyam)

[* : event] [Proc * = X # ?]]

'Proc' indicates that a sequence of actions should be returned as answer. The questions of these kinds are recognized by finding whether there is a partially or totally unknown instrumentality. The question concept of such questions is found out by removing the instrument slot. Thus for the above question, the question concept is

[Ptrans | agent^{-1} Ram | Obj^{-1} Ram | dest^{-1}(House Shyam)]

(8) Concept Completion

Concept - completion questions involve many who, where -, when - questions. These questions are something like fill-in-the -blank questions. They will give some conceptualizations with a missing component and will ask for completion of that event. Examples: 'What did Ram eat? '.

The PREP form is

[Ingest | Agent^{-1} Ram | Obj^{-1} (X # ?)]

Such questions are recognized by finding that there exists an unknown conceptual component somewhere in the comceptualization.

The question concept of such questions is same as the parsed question conceptualization.

(9) Quantification

Such questions ask for an amount. The amount may be countable. Examples of such questions are

'How many people are there? '

In PREP-expression form, this is like

[(Number(people |place^{-1} there)) : (X# ?)]

: How many cats does Ram have? '.

In PREP-expression form, this is like

[ATRANS | agent^{-1} Ram | obj^{-1} cats | dest^{-1} Ram
(Number (CATS | (* : event))) : (X# ?)]

Such questions are recognized by finding an unknown state scale value. The question concept of such questions is identical with the parsed question conceptualization.

Thus given a question we can now go through a series of analysis to find out the category of the question. This is the task of the question analyzer.

In the above categorization we showed for each question, the corresponding PREP form. '*' used in the forms is 'anaphora' in PREP and it refers to the expression in the next higher level.

e.g. [α (* : β)]

here * refers to α .

Also [α (* : event) (* : β)]

here also * refers to α in both the places.

CHAPTER 5

BANKING SYSTEM SCRIPT

In this chapter we shall discuss about script, structure and construction of the script and the way question answering is done. The domain of our question answering is the banking system. We do not consider all sorts of knowledge about banking system but only that which involves dealing with crossed cheque. The way we construct the script may give an intuitive idea of constructing scripts for other areas of question answering in the banking system.

To construct our script we consider some fictiticus characters like Ram and Shyam etc. We consider that Ram has a crossed cheque and he does not havean account in the bank under discussion. Then what he should do and how different actions are to be performed? Thus there is a main script which will use different other scripts.

MAIN SCRIPT: Withdrawal-of-money-with-crossed-cheque

preconditions are:

(a) Ram has a crossed cheque

(b) Ram does not have an account of the bank

(c) Ram knows Shyam who is an account holder of the bank.

This script will have three parts:

(I) Transfer of cheque by Ram to Shyam.

(II) Withdrawal of money by Shyam.

(III) Shyam gives money to Ram.

Thus these three parts may be made to form three different scripts with various preconditions, main acts and so on.

Now we shall start from the base i.e. we shall consider the construction of these three basic scripts and will assemble them to form the main script.

Script 1: Transfer of Cheque

Maincond: Ram knows Shyam. Shyam is account holder of bank.

FACTS : Man sitting at the counter is an officer. Colour of slip is blue.

Summary : Ram transfers cheque to Shyam's account.

Main act: Ram filled up yellow slip and gives to an officer at a counter.

Causal-Chain Representation :

Ram took yellow slip from an officer.
| I/R
Ram wrote details of cheque on the slip
| Leadto
Ram wrote account number of Shyam on the slip.
| I/R
Ram gave the slip to the officer.
| I/R
Ram gave the cheque to the officer
| Leadto
The officer put a stamp on the slip.
E |
The Officer gave back a counter-foil to Ram
| I

Ram decided to keep the foil till the money comes in Shyam's pass-book

| R

Ram kept the counter-foil for some days

| Leadto

Ram saw the money in Shyam's pass book

| I/R

Ram discarded the counter-foil.

Script 2: Withdrawal of money by Shyam.

Maincondition: Shyam has required money in his account.

Shyam has a passbook.

Summary: Shyam withdraws money from his account.

Mainact: Shyam filled up withdrawal slip.

Causal-Chaining:

Shyam took a withdrawal slip

| I/R

Shyam wrote account number on the slip.

| Leadto

Shyam wrote ledger number on the slip.

| I/R

Shyam submitted the slip at the withdrawal counter.

| Leadto

Shyam submitted passbook at the withdrawal counter.

| I/R

The man at the counter gave Shyam a token.

| Leadto

Shyam kept on waiting before the counter.

| Cancause

The man at the counter read out the token number.

| I/R

Shyam went to the counter.

| E

Shyam gave the token back.

| Leadto

Shyam got the required money.

Script 3 :

Maincondition: Shyam had required money.

Summary: Shyam gave money to Ram.

Mainact: Shyam gave money to Ram.

Causal-chain:

Shyam had required money

| Leadto

Shyam gave money to Ram

| I/R

Ram said 'thanks' to Shyam.

In conceptual-dependency form the scripts will now the traced out and after that we shall discuss the search strategies for different questions.

MAIN-SCRIPT

Maincond: [(ATRANS | agent^{-1} Ram | obj^{-1} Crossed-cheque | dest^{-1} Ram) : event]

[((account-holder bank) ; SHYAM)]

[- ((account-holder bank) : RAM)]

Summary: [(TRANSFER| agent^{-1} Ram|obj^{-1} crossed-cheque |dest^{-1} Shyam)]

[(WITHDRAW|agent^{-1} Shyam| obj^{-1} money)]

[(ATRANS| agent^{-1} Shyam|obj^{-1} money| dest^{-1} Ram)]

MAINACT: [TRANSFER|agent^{-1} Ram|obj^{-1}crossed-cheque
|dest^{-1}(account Shyam)]

MAINBODY: [(Script-name (transfer|obj^{-1} crossed-cheque))
= G0001]

[(Script-name(withdraw|obj^{-1}money)) = G0002]

[(Script-name (ATRANS|obj^{-1} money))= G0003]

Now we consider the scripts. The scripts are kept in the property list of the atoms G0001, G0002 and G0003 respectively.

Script 1:

MAINCOND: [- (ATRANS|agent^{-1} Ram|obj^{-1}(account bank)|dest^{-1} Ram)
: event]

[(Account-holder bank) : Shyam]

FACTS : [(man counter) : officer] [(colour slip): yellow]

Summary: [TRANSFER|agent^{-1} Ram|obj^{-1} crossed-cheque
|dest^{-1}(account Shyam)]

MAINACT: [WRITE |agent^{-1} Ram|obj^{-1}(egery-details)
|dest^{-1}(yellow-slip)]

Causal-chain:

[ATRANS-|agent^{-1} officer|obj^{-1}slip|dest^{-1} Ram]
| I/R
[WRITE | agent^{-1} Ram|obj^{-1}(every-details)|dest^{-1}slip]
| Leadto
[WRITE| agent^{-1} Ram |obj^{-1}(account-number Shyam)
| |dest^{-1} slip]
| I/ R
[ATRANS| agent^{-1} Ram| obj^{-1} slip | dest^{-1} officer]
| I/R

[ATRANS |agent^{-1} Ram |obj^{-1} crossed-cheque|dest^{-1}officer]
| Leadto
[Stamp |agent^{-1} Officer | dest^{-1} slip]
| E
[ATRANS | agent^{-1} officer | obj^{-1}(counter-foil)|dest^{-1}Ram]
| I
[MBUILD | agent^{-1} Ram | obj^{-1}(keep-the-counter-foil)]
| R
[ATRANS | agent^{-1} Ram | obj^{-1}(Counter-foil)|dest^{-1}Ram
| time^{-1}(somedays)]
| Leadto
[Attend | agent^{-1} Ram | obj^{-1} eyes | dest^{-1}(passbook Shyam)]
| I
[MBUILD |agent^{-1} Ram | obj^{-1}(money-is-written)]
| R
[ATRANS | agent^{-1} Ram | obj^{-1} counter-foil|dest^{-1} Ram
|time^{-1} nomore]

Script-2

Maincond: [ATRANS|agent^{-1} Shyam|Obj^{-1} passbook |dest^{-1} Shyam]
[ATRANS|agent^{-1} Shyam|obj^{-1}(required-money-to-be-given-to-Ram-in-account)|dest^{-1} Shyam]

Summary: [WITHDRAW|agent^{-1} Shyam |obj^{-1}(money account)]

MAINACT: [WRITE | agent^{-1} Shyam | obj^{-1} (every-details)
|dest^{-1} withdrawal-slip]

Causal-Chaining:

[ATRANS|agent^{-1} Shyam|obj^{-1} withdrawal-slip|dest^{-1} Shyam]
| I/R
[WRITE |agent^{-1} Shyam |obj^{-1} account-number|dest^{-1}

| I/R

[WRITE|agent^{-1} Shyam |obj^{-1} ledger-number|dest^{-1} withdrawal-slip]

| I/R

[ATRANS|agent^{-1} Shyam|obj^{-1}(withdrawal-slip) |dest^{-1} (man withdrawal-counter)]

| I/R

[ATRANS|agent^{-1} Shyam |obj^{-1} pass book|dest^{-1} (man withdrawal-counter)]

| Leadto

[ATRANS|agent^{-1}(man withdrawal-counter)|obj^{-1} token| dest^{-1} Shyam]

| Leadto

[Attend|agent^{-1} Shyam|obj^{-1} ears|dest^{-1} withdrawal-counter]

| CANCAUSE

[SPEAK|agent^{-1}(man withdrawal-counter)|obj^{-1}(tokennumber)]

| Leadto

[PTRANS|agent^{-1} Shyam |obj^{-1} Shyam |dest^{-1} withdrawal-counter]

| E

[ATRANS|agent^{-1} Shyam|obj^{-1} token|dest^{-1}(man withdrawal-counter)]

| Leadto

[ATRANS|agent^{-1} (man withdrawal-counter)|obj^{-1}(required-money)|dest^{-1} Shyam.]

Script-3:

MAINCOND: [WITHDRAW|agent^{-1} Shyam|obj^{-1} (money account)]

Summary: [ATRANS|agent^{-1} Shyam|obj^{-1}(required-money)
|dest^{-1} Ram]

MAINACT: [ATRANS|agent^{-1} Shyam|obj^{-1}(required-money)
|dest^{-1} Ram]

Causal-chain:

[ATRANS|agent^{-1} Shyam|obj^{-1} (required-money)
|dest^{-1} Shyam]

| Leadto

[ATTRANS|agent^{-1} Shyam|obj^{-1} money|dest^{-1}Ram]

| I/R

[SPEAK|agent^{-1}Ram|obj^{-1} Thanks|dest^{-1} Shyam.]

CHAPTER 6

SEARCH STRATEGIES FOR DIFFERENT QUESTIONS

The script structure gives a bird's eye-view of the whole knowledge. So for the searching of answer for each question we shall search at two levels, first at the script structure level and then at the causal-chain level. For search at the second level we need a pre-question statement to match against a script summary. We now consider the question categories one by one:

(1) Causal Antecedent:

(a) Script-structure retrieval heuristics:

We first find the question concept and try to match with the conceptualisations in the summary of the main script. There are three conceptualisations. If the question concept or main part of it matches with one of the conceptualisations, then we should invoke script-structure specific heuristics.

Now we try to match the question concept with the summary of script-1. If the match succeeds, then the MAINCOND of Script-1 is returned as answers of the questions. If it matches with the summary of script-2 then MAINCOND of script-2 is returned as answer. If the question concept matches with summary of script-3, then MAINCOND of script-3 is returned as the answer.

Causal-Chain: If the pre-question statement matches with any of the summaries then we look into the causal chaining to find the answer for the question. We try to match the question concept with a particular conceptualisation in the causal chaining and try to find if there is a 'LEADTO' link between this concept and the previous concept. If so then the previous concept is returned as the answer.

Otherwise if the link is E or I/R, then search is made for the link between previous two conceptualisations. If it is LEADTO, then the conceptualisation just before 'LEADTO' is the answer.

(2) Search for Goal-Orientation:

The search for such category of questions at the script level is obtained as follows. If the question concept matches summary of script-1, then the answer is the summary of script-2. If the question concept is summary of script-1 , then the answer is the summary of script-2. If the question concept is summary of script-2, then the answer is the summary of script-3. If the question concept matches summary of script-3, the answer will be nil.

If now the pre-question statement matches with any of the script summaries, search will be made in the causal-chaining and if inbetween two conceptualization, 'Reason' link occurs, and the question concept matches with the second conceptualisation, the first conceptualisation is returned as the answer.

Causal-Chain: If the pre-question statement matches with any of the summaries then we look into the causal chaining to find the answer for the question. We try to match the question concept with a particular conceptualisation in the causal chaining and try to find if there is a 'LEADTO' link between this concept and the previous concept. If so then the previous concept is returned as the answer.

Otherwise if the link is E or I/R, then search is made for the link between previous two conceptualisations. If it is LEADTO, then the conceptualisation just before 'LEADTO' is the answer.

(2) Search for Goal-Orientation:

The search for such category of questions at the script level is obtained as follows. If the question concept matches summary of script-1, then the answer is the summary of script-2. If the question concept is summary of script-1 , then the answer is the summary of script-2. If the question concept is summary of script-2, then the answer is the summary of script-3. If the question concept matches summary of script-3, the answer will be nil.

If now the pre-question statement matches with any of the script summaries, search will be made in the causal-chaining and if inbetween two conceptualization, 'Reason' link occurs, and the question concept matches with the second conceptualisation, the first conceptualisation is returned as the answer.

(3) Enablement Search:

Here also the search will be made first at the script structure level. If the question concept matches with any one of the summaries of the three scripts, then the answer will be MAINCOND of that script and also if the summary does not match with the MAINACT of the script, then that conceptualisation should also be returned along with MAINCOND.

If the search fails at the script structure level, then we try to match the prequestion statement with the summaries. If it matches with a summary, then we search for two conceptualisations linked by a 'ENABLE' link and if the second conceptualisation matches with the question concept, then the first one is returned as an answer.

(4) Causal Consequent Search:

This category of questions are answered in exactly the same manner as the Goal orientation search. Thus if the question concept matches with the summary of a script, the answer is the summary of the next script.

If the search fails we then search in the causal chain just in a way opposite to causal antecedent search.

(5) Verification Search:

These questions are easily answered by 'yes' or 'no'. We consider only this type of answering. Taking question concept, we try to match in the whole knowledge base. If it succeeds, then 'yes'.

(6) Disjunctive questions search:

Disjunctive questions are nothing but verification questions connected by Or's. Each verification question is taken and searched in the scripts to match. If a single match succeeds, the answer returned is 'yes'.

(7) Concept completion search:

Such questions can be also searched at the script structure level. If there is no pre-question statement, then matching is tried with MAINCOND, SUMMARY, MAINACT, FACTS of each scripts. If the matching succeeds, we return the value of the variable in question, as the answer.

Otherwise, if the prequestion statement matches with a script summary , we try to match the question concept in the causal chains.

(8) Search for instrumental questions:

Answers to instrumental questions can be found either at the script structure level or at the causal chain level.

Script Structure Retrieval: The matching search examines summaries for each script and if the question concept matches with the summary of a particular script, then the summaries of all the previous scripts are sent as answers.

If on the otherhand, the prequestion statement matches with a script summary, the search is made at the

causal chain level. The search runs through the causal chain looking for an answer key. If one is found then the answer key is checked to see whether it has an instrument slot. If so, the conceptualization is returned as the final answer.

(9) Search for feature specification/ quantification questions:

Such questions are very much like concept completion questions but their answers are found in the FACTS of a script. If the prequestion statement matches a script summary, we search in the FACTS to find an answer for the question.

Concluding Remarks:

In the above design we have considered only the script structures and not the planning structures. Also we have considered a particular domain like banking system. For some other domains the design may differ greatly.

CHAPTER 7

CONCLUSIONS

The computational model of question answering in this thesis is based on a theory of conceptual information processing using models of human memory organization. PREP has been used as a powerful tool. From the design, the following claim can be made.

Retrieval heuristics and memory representations are closely related. The question answering task provides concrete criteria for judging the strength of a memory representation.

In the above thesis we dealt with problems like memory representation, question categorization, memory retrieval, knowledge structures in memory etc. Some of the problems that we did not address are: Knowledge state assessment, use of conversational rules, conceptual recategorisation etc. The above system can be extended to cover these problems as well.

REFERENCES

1. Lehnert, W., The Process of Question Answering. LEA Publishers, New Jersey.

2. Michalek, T.F., M.Tech. Thesis (MIT 1982).

3. Schank and Abelson, Scripts Plans Goals and Understanding. LEA, New Jersey (1977).

4. Henderson, P., Functional Programming Application and Implementation. PHI, London.

5. Winograd, T., Understanding Natural Language. Newyork: Academic Press, 1972.

6. Sangal, R., Towards a Methodology for Representing Knowledge (unpublished).

7. Lehnert, W., Human and Computational Question Answering. Cognitive Science, 1977, 1(1).

8. Waltz, D., An English Language Question Answering System for a Large Relational Data Base. Urbana, I11: University of Illinois, 1977.

9. Norman, D., Memory, Knowledge and Answering of Questions. San Diego: Centre for Human Information Processing, University of California, 1972.

10. Charniak, E., A Partial Taxonomy of Knowledge About Actions. Proceedings on AI, Tbilisi, USSR, 1975.